

CLAIMS:

1. A method for re-inserting a packet, comprising:
 - determining if an audit mode is active;
 - responsive to the audit mode being active, determining if a packet has previously been partially processed by network interface circuitry disposed in a host computer system;
 - if the packet has previously been partially processed by the network interface circuitry, allowing the packet to be processed by the network interface circuitry; and
 - if the packet has not previously been partially processed by the network interface circuitry, testing the packet for network interface compatibility with the host computer system.
2. The method, according to claim 1, wherein the host computer system is programmed for testing the packet.
3. A method for handling a burst of packets sent to network interface circuitry, comprising:
 - buffering the packets in memory accessible by the network interface circuitry;
 - incrementing a counter of the network interface circuitry for each of the packets buffered;
 - checking for a connection table entry for the packets buffered;
 - responsive to non-existence of the connection table entry, sending the packets to network interface software for preparing the packets for the network interface circuitry, the network interface software for:
 - building the connection table entry;

at least partially processing the packets;
sending the packets as processed to the network interface circuitry;
forwarding the packets from the network interface circuitry;
correspondingly clearing the buffer of the packets responsive to the
packets forwarded; and
correspondingly decrementing the counter for each of the packets cleared
from the buffer.

4. The method, according to claim 3, wherein the packets are buffered in local memory of the network interface circuitry.

5. The method, according to claim 4, wherein the packets are User Datagram Protocol formatted packets.

6. The method, according to claim 4, wherein the packets are Voice Over Internet Protocol formatted packets.

7. The method, according to claim 3, further comprising generating a total count signal from the buffer, wherein the total count signal indicates to the network interface circuitry whether all the packets sent to the network interface software for processing have been at least partially processed.

8. The method, according to claim 3, further comprising setting a do not use flag for the connection table entry, wherein the packets subsequent to an initial received packet for a connection and to creation of the connection table entry are sent to the network interface software for processing responsive to the do not use flag being set.

9. The method, according to claim 3, wherein the packets are completely

processed by the network interface software.

10. The method, according to claim 3, further comprising completing the processing of the packets with the network interface circuitry.

11. An apparatus for handling a burst of packets sent to network interface circuitry, comprising:

- means for buffering the packets in memory accessible by the network interface circuitry;

- means for incrementing a counter of the network interface circuitry for each of the packets buffered;

- means for checking for a connection table entry for the packets buffered;

- means for sending the packets to network interface software for preparation for the network interface circuitry responsive to one of non-existence of the connection table entry and a do not use flag, the network interface software including:

- means for building the connection table entry;

- means for at least partially processing the packets in sequence; and

- means for sending the packets as processed to the network

- interface circuitry;

- means for forwarding the packets as at least partially processed from the network interface circuitry;

- means for clearing the buffer of the packets respectively responsive to the packets forwarded; and

- means for decrementing the counter respectively for each of the packets cleared from the buffer.

12. The apparatus, according to claim 11, wherein the memory is local memory of the network interface circuitry.

13. The apparatus, according to claim 11, wherein the counter is coupled to the memory.

14. The apparatus, according to claim 11, further comprising means for generating a total count signal from the buffer, wherein the total count signal indicates to the network interface circuitry whether all the packets sent to the network interface software have cycled out.

15. A signal-bearing medium containing a program which, when executed by a network interface, causes execution of a method comprising:

- buffering packets received in memory accessible by network interface circuitry;

- incrementing a counter of the network interface circuitry for each of the packets buffered;

- checking for a connection table entry for the packets buffered;

- responsive to non-existence of the connection table entry, sending the packets to network interface software for preparing the packets for the network interface circuitry, the network interface software for:

- building the connection table entry;

- at least partially processing the packets; and

- sending the packets as at least partially processed to the network interface circuitry;

- forwarding the packets as at least partially processed from the network interface circuitry;

- correspondingly clearing the buffer of the packets responsive to the packets forwarded; and

- correspondingly decrementing the counter for each of the packets cleared from the buffer.

16. The method, according to claim 15, further comprising setting a do not use flag for the connection table entry, wherein the packets obtained after creation of the connection table entry are sent to the network interface software responsive to the do not use flag being set.

17. The method, according to claim 15, wherein the packets are completely processed by the network interface software.

18. A system for handling a burst of packets, the system comprising:
a central processing unit;
system memory coupled to the central processing unit; and
a network interface coupled to the system memory and the central processing unit, the network interface configured to:
buffer packets in memory accessible by a circuitry portion of the network interface;
increment a counter of the circuitry portion for each of the packets buffered;
check for a connection table entry for the packets buffered;
send the packets to a software portion of the network interface responsive to non-existence of the connection table entry, the packets sent to prepare the packets for the network interface circuitry, the software portion configured to:
build the connection table entry;
at least partially process the packets; and
send the packets as at least partially processed to the network interface circuitry;
forward the packets as at least partially processed from the network interface circuitry;

correspondingly clear the buffer of the packets responsive to the packets forwarded; and

correspondingly decrement the counter for each of the packets cleared from the buffer.

19. The system, according to claim 18, further comprising an input/output interface located between the central processing unit and the circuitry portion.

20. The system, according to claim 18, wherein the circuitry portion comprises a network processing unit.